软件引用及其规范的理论探索和实践进展

■ 丁敬达　郑巧　刘超
上海大学图书情报档案系　上海 200444

摘　要：[目的/意义] 梳理软件引用及其规范的理论和实践现状，分析存在的困难和问题，促进软件引用规范和标准的建立。[方法/过程] 通过文献调研，得出软件引用面临接受文化、奖励制度、引用认知和元数据 4 个方面挑战，分析国内外为应对挑战对软件引用及其规范的理论探索和实践进展。[结果/结论] 软件引用利益相关者分析、软件引用原则和元数据标准等理论探索以及国际研究组织和社区的引用指南、相关项目和知识库的支持、人员培训、贡献分配和奖励计划的实施、软件引用文件的提供等实践为软件引用规范的建立奠定了良好的基础，但尚需软件引用利益相关者通力协作、共同克服面对的困难和挑战。

关键词：软件引用　软件引用原则　软件引用规范　软件引用实践
分类号：G253
DOI：10.13266/j.issn.0252－3116.2021.11.015

1　引言

数字学术环境使科研需求从传统文献信息逐渐扩大到数据、软件、程序代码等电子资源。软件不仅是科研活动顺利实施的基础和保障，也是科研产出的重要成果形式之一，其通过对数据的采集、处理、分析等，使得科研结果具有可再现性和可重复性，提高了科学研究的可靠性和效率。软件作为电脑中的非有形部分，是一系列按照特定顺序组织的电脑数据和指令[1]。在科学活动中，软件主要指科学研究中用于产生科研结果的软件，但不包含操作系统、接口、数据库、算法以及数据结构等[2]。依照学术规范，软件引用是科研人员将产生科研结果的软件罗列在出版物的参考文献中[3]，著者在撰写或编撰论著时所使用的软件应被引用。软件作为一个动态的、不断发展的知识实体，在其生命周期中通常有不同的版本发行、漏洞修复等[4]，软件固定性和粒度使得软件引用的学术规范研究进展缓慢，迄今还未形成统一的、国际性的规范和标准。而国外软件引用的理论探索与实践早于国内，并已取得一些进展，国内对软件引用整体研究时间较短，定量分析为主要研究方向，且研究处于分析软件引用的特征及其问题的阶段，理论和实践层面尚未形成统一的规范和有效的评价体系。笔者以 Web of Science、Scopus、CNKI、Bing 等作为文献检索源，从软件引用的定量研究出发，首先归纳软件引用的挑战，然后对国内外应对挑战的理论探索及实践进展进行梳理，分析存在的主要困难和问题，促进软件引用规范和标准的健康发展。

2　软件引用的定量研究

软件引用的定量研究主要从科研人员和软件两个视角出发，采用小样本的人工编码或大样本的机器学习方法，在数据库或社区中对软件引用的特征及其存在的问题进行统计和描述。

2.1　软件引用行为

软件引用行为分析主要立足于不同学科的科研人员视角，通过全文本数据库或引文数据库对科研人员引用当前软件的行为进行分析。从研究范围来看，按照学科领域划分，国内外科研人员的研究范围主要涵盖了图书情报学、生物信息学、地球动力学、天文学以及综合性学科。从研究方法来看，不同领域通常采用内容分析法作为主要研究方法来分析科研人员的软件引用行为特征，但相较于传统人工的内容分析法的耗时费力，研究方法更加多元，融入了数据挖掘技术，数据量也明显增多。从研究思路来看，图书情报学领域[5]重在分析该领域期刊论文中软件是如何引用以及科研人员引用实践的一致性；生物信息学领域[6-7]主

作者简介：丁敬达（ORCID：0000－0002－4075－8604），教授，博士，E-mail：djdhyn@126.com；郑巧（ORCID：0000－0003－0684－9884），硕士研究生；刘超（ORCID：0000－0002－9333－7838），博士研究生。
收稿日期：2020－12－28　修回日期：2021－03－01　本文起止页码：143－152　本文责任编辑：徐健

要通过对比中外文献研究，来分析软件引用特征、软件和数据集对论文的影响力等；地球动力学领域[8]结合了定性与定量方法评估了软件开发和归属的当前实践，以及对软件引用的需求和最佳实践的态度；天文学领域[9]以某一特定软件作为研究对象来探讨该领域的引用实践情况；综合性学科[10]重在通过将学科进行分类，对比分析各学科科研人员的引用行为。

从学科领域来看目前的研究范围仍具有局限，研究涵盖范围不够广泛，且由于学科性质差异，各学科科研人员使用软件的频率也不尽相同，理工科更多因研究需要而使用与引用软件。此外，同一学科也会因为论文类型不同导致软件引用与使用行为不同，如综述类论文与数据挖掘与分析类论文。通过对不同学科的研究结论进行分析，可以发现研究思路的不同导致研究结论有些差异，但这些结论主要围绕软件引用存在系列问题展开，集中体现在软件接受度低、引用实践和标准缺乏、学术奖励制度不完善等问题[11-12]，不同研究范围以及不同方法下的软件引用现状均反映出软件引用的缺失和引用实践并未改善的现状。

2.2 软件引用计量

软件的引用计量主要立足于软件本身，依据研究对象划分，软件角度的研究分为两类：一类是以软件功能划分，以某一类功能的系列软件作为研究对象，如知识图谱软件、文献计量学软件等系列软件。这类研究会结合测量指标以及一些研究方法综合展开，如扩散宽度、扩散时间以及扩散速度等知识扩散测量指标[13-14]来探讨软件引用以及扩散过程。另一类是以特定软件作为研究对象，主要包括该特定软件及其相应的软件包，如开源软件 Word2Vec[15]、R 及其包[16-18]等。

依据研究方法划分，软件的引用计量与替代计量、自然语言处理等相结合，通过新方法或新视角来获取软件的下载量、引用量等影响力相关数据，从而测量软件在科研活动中的影响程度。关于新方法或新视角的软件引用研究主要包括 Altmetrics 视角以及软件实体识别与抽取的方法。赵蓉英等[19-20]从 Altmetrices 角度出发，构建下载量、被引指标、软件复用指标，并对 Python 社区的部分软件进行学术影响力评价。但 Altmetrics 视角以及软件实体识别与抽取方法的侧重点不同，Altmetrics 更侧重于测评软件的学术影响力，软件实体识别与抽取主要从技术方法出发，从方法层面了解目前相关技术在软件引用研究上的使用情况，因此软件实体识别与抽取则更注重软件使用与引用的分布特征，如 X. Pan 等[21]提出了一种在全文本论文中抽取软件实体的改进的 boostrapped 方法，通过与 Basilisk、NOMEN 和 SPIED3 个实体提取系统进行对比，发现软件引用存在二八定律，学术免费软件比专利软件更易获得引用。

3 软件引用面临的挑战

通过对软件引用现状的分析可以看出，软件引用主要面临软件接受文化、软件奖励制度、软件引用认知和软件引用元数据 4 个方面的困难和挑战。

3.1 软件接受文化

在软件引用中，最具挑战性的是非技术层面的问题。软件是现代科学研究的重要组成部分，其对科学研究中的数据分析、结果呈现、知识传播等方面都具有重要作用，研究也表明科研人员高度肯定了软件在科研工作中的重要性[22-24]。但软件的文化接受度较低，其被视为辅助工作而非科研成果，只是支撑科学研究的工具且为科研服务[25]，整个学术生态系统对其发表、确认和引用都缺乏支持[26]。D. S. Chawla 指出在现有科研环境下，存在一种文化使科研人员认为与撰写论文相比，编写代码并非为主要研究目的[27]，这表现为科学、工程和许多其他学术领域仍然将科研论文视为科研生产力的关键指标，且科研人员很少将代码或软件列为其科研产出之一。此文化观点在 L. Hwang 等人的研究中亦有体现，该研究中受访者认为编写软件并不是一门科学，开发软件所需的时间和精力被认为是发表科研成果的障碍，即开发软件的研究效率不如同行评议的出版物[28]。而研究软件作为一项创造性的工作，应该得到承认和引用。因此，不论是研究团体还是推广委员会、资助者和出版商，其对软件的文化态度需要改善，将软件视为有价值的科研成果产出，否则缺少软件引用有可能导致知识丢失和资源浪费[29]。

3.2 软件奖励制度

虽然培育软件的接受文化是软件引用中一项重大挑战，但文化与制度相辅相成，相互影响，软件引用的低文化接受度折射出其背后的学术奖励制度不够完善。传统的学术奖励制度是基于出版物标准和科研影响，而软件是非传统的文献形式，在实践中，未包含在同行审查的出版制度中，故科研人员在软件方面的贡献被掩盖[28]。由于出版物拥有比软件开发更高的评级和声誉，外加软件缺乏引用的实际情况，软件开发人员可能无法获得与发表备受推崇的期刊论文同等的学

术声誉[30]。J. Howison 等[31]提出科研人员更看重声誉,然而学术奖励制度存在激励不足的问题,在以出版物为主的学术声誉中,科研人员若通过开发软件来建立独立的学术声誉需要付出巨大的额外努力,同时软件实践可能会降低软件开发者的优势来进一步影响其获得学术声誉。这对于从事科研的工作人员是一种惩罚,所以应该承认新形式的学术奖励,并随着新方法和新技术的发展来扩大和发展学术信誉分配[32]。

3.3 软件引用认知

与热门的科学出版物不同,软件及软件出版物本身很少被引用,软件引用实践的多样性表明出版物对软件引用规范要求不太明确,一方面并非所有软件用户都认为需要引用软件,另一方面并非所有软件包都提供方便的引用,即缺少引用格式,这使得用户不清楚应该引用谁以及如何正确引用[31]。在研究过程中,用户对于引用软件存在两种争议:一种观点认为引用所有使用过的软件;另一种认为引用仅仅对研究结果有意义的软件。第一种观点使得软件和科研结果变得绝对和清晰,第二种观点由软件用户判断决定,变得相对模糊[32]。软件作为可引用对象,其信息不一致而导致的大量变体存在以及缺乏整合,对使用引用数据去追踪软件实体的科学影响力评价造成很大的挑战[33]。事实上,很难有某种引用方法能够完全满足科学研究重现的需要,因此对如何引用和参考软件以及如何构建合理的软件引用规范体系缺乏一致的认知[34]。引用标准的不一致使得为软件实现任何单一引用框架的工作变得复杂,然而即使采用了软件引用的标准方法,索引数据库(如 Web of Science、Scopus、谷歌学术等)目前也缺乏对此类引用的索引支持[35]。

3.4 软件引用元数据

随着科学数据的兴起,研究粒度逐渐细化到数据、软件等知识实体单元[36－38],科学数据引用的元数据逐渐引起重视。软件作为科学数据,其引用实践没有提供足够的元数据来支持软件引用的功能,引用信息的缺失不仅仅表现为引用内容的缺失,还包括软件名称、版本等。一方面是因为缺乏元数据标准[39];另一方面是由于软件的特性,如软件版本信息更新,不同版本有不同的贡献者等,导致软件引用率没有准确反映使用率[40]。同时出于不同研究目的,研究人员从软件中获取的元数据也并不一致。例如软件声誉分配获取的元数据主要为作者、标题等;软件和数据的可再现性或可重复性侧重软件版本等信息;研究软件的可发现性则更关注的是软件的关键词和描述等[41]。H. Park 等[30]通过对软件元数据字段进行统计分析,发现要实现精确引用来使软件有效表示仍然具有很大的挑战。软件开发和软件信息的广泛分布,使软件与现有基础设施的适当集成变得复杂[42]。不同的软件存储库、软件语言和科学领域以不同的方式表示信息,这种异源异构信息的集成工作使得许多有价值的元数据在集成过程中丢失。在科学数据仓储元数据标准方面,DataCite 和 Dataverse 提供了引用标准和格式,但 DATS、W3C HCLS Dataset Description 和 Dryad 科学数据仓储并未提供元数据标准[43]。目前,软件引用元数据标准缺乏,从科学元数据标准来看,应用比较广泛的科学元数据标准为学科领域元数据标准和通用型元数据标准,学科领域的元数据标准发展不均衡,且实用性不强[44];在通用型元数据标准中,仅有都柏林核心元素集在科学数据管理中被应用,但其并不能完全覆盖科学数据管理周期的所有要求,通用型元数据标准并未得到统一制定与使用[45]。

4 软件引用及其规范的理论探索

软件引用及其规范的理论研究主要从利益相关者、软件引用原则、元数据标准 3 个方面展开。在理论层面,软件引用涉及众多利益相关者,需不同利益相关者的密切合作,软件引用原则、元数据标准的研究主要为软件引用提供相应的理论依据和实践指导。

4.1 软件引用利益相关者

软件引用是一项复杂性工程,其存在的问题没有一个利益相关者能够独立解决,需要各利益相关者之间密切合作,明确各利益相关者的角色与责任,从而以点带面地形成软件引用接受文化。在具体针对软件引用利益相关者分析方面,科研人员提出了不同的分类标准和协作方案。D. S. Katz 等[46]认为利益相关者包括使用软件的特定领域科研人员社区、科研软件工程师和其他软件开发人员、科研机构、出版商、存储库提供商、索引提供商和资助者。基于该分类,S. Druskat[47]提出了不同利益相关者采取行动的流程。E. Henneken 等[48－50]在 Asclepias 项目中建立了出版商、存储库、索引服务商 3 个利益相关者协作的框架。A. Allen 等[51]从软件开发人员、软件使用者、期刊编辑、社区角度提出协作方案。在科学数据开放共享的环境下,软件利益相关者分析的范围也涵盖于科学数据的利益相关者分析之中。盛小平等[52]分析了政府、研究人员、研究机构、研究资助机构、图书馆或档案馆及数据中心、出版商、专业协会或学会、用户与企业等

利益相关者的责任与义务;张丽丽等[53]从引用的预研、创建及实践出发,将利益相关者划分为科研传播机构、科研监管部门及数据用户;孟祥保[54]则从数据生命周期角度将利益相关者概括为科研人员、科研资助者、科研数据组织、科研机构、数据中心、图书馆和学术出版者。科学数据利益相关者虽然有部分适用于软件引用的利益相关者,但具体针对软件而言,也只有部分具有参考价值。

4.2 软件引用原则

软件引用的现状分析表明期刊文献中使用的软件仅小部分被引用,且格式迥异,有参考引用用户指南、网页网址等,引用位置也不尽相同,在文献的脚注、参考文献、致谢、图表等位置均有标引。由于缺乏公认的引用标准导致大多数软件都未被引用,软件的价值以及开发人员的贡献得不到发现和认可,因此规范化软件的引用显得尤为重要。但由于软件固定性和细粒度的挑战,软件引用的学术规范研究进展缓慢。

针对软件引用规范方面缺乏统一的、国际性的引用标准的问题,FORCE11 软件引用工作组[55]在理论层面上提出了软件引用原则(Software Citation Principles, SCPs),以期通过一套统一的引用原则来鼓励不同学科广泛采取一致的软件引用策略。软件引用原则是从数据引用原则的 8 项原则中选择 6 项更名而来,这些原则具体包括:重要性、归属性、唯一标识性、持久性、可访问性和特异性。其中,重要性指软件应被视为合法的、可引用的科研产品,软件的引用应与其他科研产品(如出版物和数据)的引用同等重视;归属性是指软件引用应促进学术信誉分配给软件所有贡献者,单一风格或机制可能不适用于所有软件;唯一标识性是指软件引用应包含机器可操作的、全球唯一的、可互操作的、能被识别的标识方法;持久性是指唯一标识符和元数据的生命周期应超过软件的生命周期;可访问性是指支持对软件本身及其相关元数据、文档、数据和其他材料的访问;特异性是指支持识别和访问所使用软件的特定版本。

然而软件引用原则虽然给出了原则,但并未讨论这些原则的实现[42]。在现实情况下,软件接受文化以及用户引用认知导致原则 1 的软件重要性难以得到重视;引用未提供足够的信息来支持软件作者的信誉分配,原则 2 的归属性实现受阻;原则 3、原则 4 要求的持久的、可操作的机器标识符仍未能供给,且开发者及贡献者的身份缺乏永久识别,要实现引用和评估影响力措施需要将相关数据存储在适当的保管环境中,这对现有的基础设施提出了较高的要求;且目前对于软件的引用多为 URL,原则 5 的可访问性方面不够稳定;软件版本这一主要的描述性元数据元素可能因标准缺乏、科研人员引用认知不一致等问题而缺失,因此原则 6 在实践中也具有不完全性和不确定性。

软件引用原则对于规范出版物中引用软件的适当格式具有重要作用。虽然软件引用原则并未定义应该引用什么软件,但是其定义了应该如何引用软件。软件引用原则提出了建议软件引用的格式说明:文献类型标识为[软件],若包含更多的信息,需根据信息类型标明,比如[软件:源代码],[软件:可执行文件]等;同时也需要对软件版本信息进行标明,例如,版本 1.8.7。基于软件引用原则提出的建议,A. Heinle 等[4]给出了软件引用的完整格式:标题[软件:种类(源代码、包、可执行文件、库、其他)][版本或 URL/更新日期/下载日期]。但实践中各社区并未形成统一的引用格式,部分关注软件引用的期刊提供的关于软件引用的声明或标准格式也具有差异性,形成统一或标准的软件引用格式仍道阻且长。

4.3 软件引用元数据标准

软件引用原则的提出对软件引用规范具有重要意义,形成统一或标准的软件引用格式意味着需要识别与引用软件相关的必要元数据。为使软件可被引用,其必须是可持续性、可发现和可重用的。这要求软件开发者以及贡献者提供有关软件的附加信息或元数据。而目前国内外关于软件引用元数据标准方面的研究还较少。现有研究多集中于软件引用的上一层次,科学数据引用过程中的元数据标准,覆盖范围有科学数据仓储元数据标准[43]及实践[56]、元数据编写规范及元数据标准[57]、通用型科学元数据标准研究[45]与设计[58]、标准体系[59]、国外标准研究其启示等[60]。虽不具有针对性,但具有一定的参考和借鉴价值。

可持续性对元数据中的标识符提出了要求,软件产品的唯一标识可以通过提供 DOI 或类似持久标识的归档文件来发布。DOI 作为一种独特的持久性标识符,可用于期刊文章、研究数据和软件等的跟踪引用[34]。通常情况下,现有的特定数据集及软件具有的持久标识符为 DOI,而不同于采用 DOI,2014 年生物医学领域引入了 RRIDs(Research Resource Identifiers),以期通过单一的统一标准来识别和跟踪科研论文中研究资源的使用。RRIDs 作为资源识别符在粒度等方面符合软件引用原则,并于 2019 年被纳入期刊论文标记套件,这表明学术出版界已接受其作为标记研究资源的

丁敬达，郑巧，刘超. 软件引用及其规范的理论探索和实践进展[J]. 图书情报工作,2021,65(11):143－152.

标准方法[61]。而针对软件引用中的贡献者和机构进行唯一标识的问题,技术层面上可以通过使用 ORCID (Open Researcher and Contributor Identifier)标识符来实现。ORCID 作为一种在许多不同系统中识别研究人员的方法,通过 ORCID ID 将其与科研成果相关联,可以追踪 ID 对应下的学术影响力,但 ORCID 的发展仍处于早期阶段[62],还不够完善和系统。

可发现性意味着在软件出版、引用过程中可以帮助其他人找到软件的相关信息,因此元数据最基本的元素是对找到该软件有实际帮助的元素,即标题/软件名、作者、日期、位置、标识符。K. E. Niemeyer 等[35]提出元数据应包括软件名称、主要作者/贡献者(名称和 ORCID)、DOI 或其他唯一且持久的标识符,以及软件已发布/存档的位置,若没有 DOI,则元数据应该包括定位软件的方法,如 URL 等。S. Druskat[63]从技术解决方案上开发软件元数据格式,提出了基于 YAML 1.2 的人类和机器可读的引用文件格式(Citation File Format,CFF),指出软件引用元数据应包含摘要、作者、提交信息、相关联系人、发布日期、DOI、关键词、授权信息、存储库、标题、URL、版本等。另外,元数据信息应该以 citation 文件的形式提供,元数据形式可以是 JSON 或 XML 格式以及适当的元数据模式,例如 DOAP,从而允许自动处理[35]。

软件引用元数据标准更多需要的是技术方面的支持,但现状是软件出版实践不尽如人意,缺乏正确完整的元数据,以及对软件引用的激励和要求。不过即使解决了技术方面的问题,元数据采用仍然是一个文化挑战,可以通过鼓励、请求和要求采用来自资助者、出版商和机构的标识符,以及通过研究人员、软件创建者和机构的教育和示范实践来解决[47]。

5 软件引用的实践进展

针对软件引用的挑战,软件引用的实践进展主要包含国际组织研讨会和社区指南、技术支持的项目及知识库、人员培训、实施贡献分配和奖励计划、提供软件引用元数据。软件引用作为一种学术规范,学科共同体应当形成公认的认知,遵循共同的具有约束力的学术道德和学术规范,这不仅需要对技术上进行规定,也需要在道德上形成约束和共识。

5.1 国际组织研讨会和社区指南

近年来,软件引用存在的问题逐渐引起学术界的关注,一些国际组织或机构陆续开展了有关软件引用的研讨会。WSSSPE(Working towards Sustainable Software for Science:Practices and Experiences)作为一个由社区推动的国际组织,主要通过共享学习和社区行动来促进软件的可持续发展,该组织已陆续开展了国际研讨会,如 WSSSPE 5.1、WSSSPE 5.2、WSSSPE 6.1 会议等[64]。FORCE 11 是通过信息技术来推动和改进知识创造和共享的社区,其成立了许多工作小组来开展软件引用相关的项目,如研究引用中的贡献分配归属的工作组[65]、研究软件识别的 RDA 联合软件源代码识别工作组[66]、软件引用原则工作组[67]、制定实施原则的指导方针并助力实现原则的软件引用实施工作组[68]等。软件可持续性研究所(Software Sustainability Institute, SSI)[69]举办了软件可持续性研讨会 WOSSS19、软件存放与保存研讨会、软件信誉分配研讨会等。

软件引用原则的发布为软件引用规范奠定了良好的基础,部分社区及期刊纷纷建立了社区指南及期刊声明或标准,并改进工具和基础设施来支持软件引用。天体物理学社区在美国天文学会(American Astronomical Society,AAS)期刊上给出了引用描述软件的论文或软件 DOI 两种软件引用标准[70];数学社区开发和改进了 swMATH[71];科学技术设施委员会(the Science and Technology Facilities Council,STFC)提供了 HSL 软件索引的推荐引用格式[72];生命科学社区给出了构建美国国立卫生研究院(The National Institutes of Health,NIH)生物医学软件发现索引的指南[73];此外,包括软件本身和软件论文的软件出版物的引用也逐渐引起重视,如 F1000 研究、生物信息学、计算机物理通信等逐渐接受并发表软件论文,但该类期刊要么接受软件论文作为一种特殊的提交类型,要么只考虑针对该领域的软件论文;而另一类只考虑软件或软件论文的提交而不关注特定领域的开源期刊也陆续创建,如 JOSS[74]、JORS[75]、SoftwareX[76],这类期刊在相应的网站政策中都提供了关于软件引用的声明或标准。

5.2 相关项目和知识库的支持

一些关于软件引用的项目也在逐渐进行中,这些项目关注软件引用过程中涉及到的不同主题。CodeMeta 项目[77]主要聚焦元数据,通过一个 crosswalk 表和一个最小的软件元数据模式,在软件库、注册中心以及归档所使用的元数据字段之间进行映射,提供跨软件元数据格式的兼容性信息,并用 JSON-LD 实现;Shouldacite[78]关注引用规范,该项目由协作研讨会(Collaborations Workshop 16,CW16)[79]创建,其提供了关于引用特定软件的指导以及关于如何引用和获得引

用的进一步指南;Depsy 项目[80]致力于跟踪软件影响,并通过计算每个参与者贡献或编写提交代码的百分比来对每个贡献软件包的参与者分配分数;Impactstory 项目[81]将使用一个深度学习系统来建立大型数据库,并使用数据库来构建和开发 3 个工具:CiteSuggest 可以分析提交的文本或代码,使用软件作者的首选引用并对规范化引用提出建议;CiteMeAs 可以帮助软件生产商明确要求他们的首选引用;Software Impactstory 可以帮助软件作者在文献中展示其软件的学术影响。

随着开放科学的快速发展,越来越多的知识库在软件引用方面提供了技术层面的支持,如 GitHub、Zenodo、Figshare、DateCite、OntoSoft 等。通常软件首先存储在代码库 GitHub[82]中,从 GitHub 中可以自动地将一个副本存档到学术型的存储库中,如能提供 DOI 的数字存储库 Zenodo 和 Figshare,其将 DOI 分配给软件,从而使软件成为一个可引用的对象。DataCite[83]是所有存储库的中心 DOI 提供者,因此若要分配 DOI,Figshare 必须将有关对象的元数据传递给 DataCite,即软件元数据在 DataCite 注册。OntoSoft 是一种用于描述软件元数据的本体,可用于识别、理解和评估软件、执行软件、获得软件支持、使用软件进行研究以及更新软件,OntoSoft 对非软件开发人员的科学人员发现和重用元数据具有重要的作用[84]。CiteAs[85]则可从不同类型研究产品(包括软件)中来检索引用信息。opencitation 将引用的概念作为一级数据实体应用,创建了第一个开放引用索引 COCI[86]。此外,各知识库在发挥各自功能的同时,相互之间亦进行协作,共同开发新技术和新方法,为解决软件引用中存在的问题提供技术支持,如 Github、Figshare 和 Mozilla 科学实验室实现了直接链接到开发团队的链接来追踪其开发的源代码存储库以及自动化 DOI 的方法[87]。

5.3 人员培训

对于软件开发人员和科研人员而言,改善软件使用与引用的实践差异,其有效的途径之一为人员培训。社区、机构等通过制定一系列培训计划,形成对软件的态度以及处理软件方式的统一,来规范化研究人员对开发、使用及引用软件的行为。人员培训有助于规范化软件使用与引用规范,从而在思想和文化层面提高软件使用与引用的文化接受度。软件木工(software carpentry)培训计划是解决软件开发技能差距的有效工具。软件木工通过组织研讨会和课程对研究人员进行培训,能有效推动科学环境中研究人员使用、开发软件的规范[88-89]。工程与物理科学研究委员会(The Engineering and Physical Sciences Research Council,EPSRC)通过投资于研究和研究人员培训,来建立解决国家面临的科学和技术挑战所需的知识和技能基础[90-91]。

5.4 实施贡献分配和奖励计划

在实现贡献分配方面,已推出贡献者角色分类标准(Contributor Roles Taxonomy,CRediT)来识别软件开发人员等相关贡献,并在其贡献要素中对软件开发者的相关贡献进行认可。在 CRediT 推出的 14 个贡献者角色中,明确提出了软件(software)这一贡献要素,其包括编程与软件开发、计算机程序设计、计算机代码和支持算法的实施、现有程序组件测试等[92]。而软件开发不仅涉及软件开发人员,还涉及到后期的贡献者、测试者等,CRediT 虽可以识别出贡献角色,但并未能衡量每个贡献要素的贡献大小。针对软件贡献研究人员的具体贡献及贡献大小,J. Howison 等[93]提出为软件贡献研究人员提供额外的资源,以便其提出贡献的理由,并对贡献进行分类,同时科学政策制定者与期刊、会议和专业协会合作,改进出版物中软件的引用,跟踪软件代码的实际运行,收集软件的使用数据,并提供一组引用供论文使用,将数据链接到出版物并公开集成贡献。

在实施奖励计划方面,奖励计划(fellowship program)向许多跨领域、机构和职业阶段的研究人员提供奖金,来换取其专业知识和建议。奖励计划促使研究人员建立了敏锐的群体认同感、愿景和一系列活动,将软件不同利益相关者联系在一起,来探讨软件引用、使用及开发过程中的重要思想及内容。许多研究人员超越了最初的职权范围,主动在各自社区内组织活动,发表与研究有关的演讲,倡导最佳做法[94-95]。英国研究软件工程学会(the UK Society of Research Software Engineering,UKRSE)发起了一项运动,以表彰从事研究的软件工程师,为减少对出版认可的依赖性,已讨论开发一个"软件奖"。该项计划于 2015 年受到英国工程和物理科学研究委员会(Engineering and Physical Sciences Research Council,EPSRC)的资助,但需要经过严格的同行评审过程,这是全球第一个此类有此筛选机制的奖金计划[96]。美国也推出了"更好的科学软件"奖金计划(Better Scientific Software,BSSw)[97-98]。

5.5 提供软件引用文件

软件引用文件(CITATION File)是包含如何正确引用软件信息的计算机文件,其包含的引用信息对撰写论文的科研软件用户来说至关重要。提高软件发现

的一种策略是让软件创建者在 CITATION 文件中提供与引用相关的元数据,R. Wilson[99]提出的这种简单灵活并能广泛应用于科学软件的方法,需要给出引用的纯文本或 BibTex 条目,以便用户运行时打印出适当的引用,从而指导用户引用。但在高度自动化的软件引用工作流程中,需要采用机器可读的标准格式来替换自由格式的纯文本文件[100]。S. Druskat[101]主张 CITATION 文件应采用标准化格式(Citation File Format, CFF)来简化重复使用。CFF 作为 YAML 1.2 中的一种人类可读的文件格式,在技术解决方案上为软件提供了引用元数据,也开发了很多 CFF 工具用于创建、操作和转换[102]。CFF 具有很多的必填字段,可以实现软件引用原则,也可以与 CodeMeta 的 JSON-LD 等其他元数据格式兼容,CFF 在推动和实现软件引用学术规范方面发挥了重要作用。为推动 CFF 在软件引用过程中的应用,荷兰电子科学中心(Netherlands eScience Center)推出了研究软件目录来提供软件引用元数据,以此来改善研究软件的可发现性、可引用性和可再现性[103]。

6 结语

在数字科研环境下,软件引用已逐渐成为多学科领域共同关注的热点问题,软件引用不仅可以支持软件的访问、共享和再利用,而且可以提供验证科研过程的途径,并使软件开发者的贡献得到认可。笔者对软件引用及其规范化的理论与实践进行归纳、总结,认为软件引用不同利益相关者的密切合作、软件引用原则以及软件引用元数据标准等理论以及国际研究组织和社区的引用规范声明与指南、相关项目和知识库的支持、人员培训、贡献分配和奖励计划的实施、软件引用文件的提供等实践为软件引用规范的建立奠定了良好基础。但目前在其应用中还面临着软件接受度低、软件奖励制度缺失、软件引用认知不一致以及软件引用元数据标准缺乏等诸多问题和困难,需要各利益相关主体通力合作、共同推进软件引用的规范化进程。

参考文献:

[1] 软件[EB/OL]. [2020－12－15]. https://bk.tw.lvfukeji.com/baike－软件.

[2] 潘雪莲. 软件实体的自动抽取和学术影响力研究[D]. 南京:南京大学,2016:9.

[3] 中国国家标准化管理委员会. 信息与文献 参考文献著录规则 GB/T 7714－2015[S/OL]. [2020－06－03]. http://www.gs.fudan.edu.cn/_upload/article/files/e2/3e/0197d0d64cbea9d9fae9a668329f/b9a3a4c2-1e6f-4e81-9974-36eb01635471.pdf.

[4] HEINLE A, KOEPF W, SPERBER W. Some steps to improve software information [J]. ACM communications in computer algebra, 2017, 51(1): 1－11.

[5] 崔明,潘雪莲,华薇娜. 我国图书情报领域的软件使用和引用研究[J]. 中国图书馆学报, 2018, 44(3): 66－78.

[6] 杨波,王雪,佘曾溧. 生物信息学文献中的软件利用行为研究[J]. 情报学报, 2016, 35(11): 1140－1147.

[7] 杨波,王雪,苏娜. 不同文献集中中国学者引用软件和数据集的特征比较研究[J]. 图书情报工作, 2017, 61(14): 109－115.

[8] HWANG L, FISH A, SOITO L, et al. Software and the scientist: Coding and citation practices in geodynamics [J]. Earth and space science, 2017, 4(11): 670－680.

[9] WOFFORD M F, BOSCOE B M, BORGMAN C L, et al. Jupyter notebooks as discovery mechanisms for open science: citation practices in the astronomy community [J]. Computing in science & engineering,2020, 22(1): 5－15.

[10] PAN X,YAN E,HUA W. Disciplinary differences of software use and impact in scientific literature [J]. Scientometrics, 2016, 109(3): 1593－1610.

[11] PAN X,YAN E,CUI M, et al. How important is software to library and information science research? a content analysis of full-text publications [J]. Journal of informetrics, 2019, 13(1): 397－406.

[12] HOWISON J, BULLARD J. Software in the scientific literature: Problems with seeing, finding, and using software mentioned in the biology literature [J]. Journal of the Association for Information Science and Technology, 2016, 67(9): 2137－2155.

[13] 于晓彤,潘雪莲,华薇娜. 知识图谱研究中软件引用和扩散分析[J]. 情报资料工作, 2019, 40(2): 19－29.

[14] PAN X, YAN E, CUI M, et al. Examining the usage, citation, and diffusion patterns of bibliometric mapping software: a comparative study of three tools [J]. Journal of informetrics,2018, 12(2):481－493.

[15] 于晓彤,潘雪莲,华薇娜. 基于内容分析法的开源软件使用与引用研究[J]. 图书情报研究, 2020, 13(1): 101－107,121.

[16] LI K, CHEN P-Y, YAN E. Challenges of measuring software impact through citations: an examination of the lme4 R package [J]. Journal of informetrics, 2019, 13(1): 449－461.

[17] LI K, YAN E, FENG Y. How is R cited in research outputs? Structure, impacts, and citation standard [J]. Journal of informetrics, 2017, 11(4): 989－1002.

[18] LI K, YAN E. Co-mention network of R packages: Scientific impact and clustering structure [J]. Journal of informatrics, 2018, 12(1): 87－100.

[19] 赵蓉英,魏明坤,汪少震. 基于 Altmetrics 的开源软件学术影响力评价研究[J]. 中国图书馆学报, 2017, 43(2): 80－95.

[20] ZHAO R, WEI M. Impact evaluation of open source software: an altmetrics perspective [J]. Scientometrics, 2017(110): 1017－1033.

[21] PAN X, YAN E, WANG Q, et al. Assessing the impact of software on science: a bootstrapped learning of software entities in full-text papers [J]. Journal of informetrics, 2015, 9(4): 860-871.

[22] SILVA R. Measuring behaviour 2018 [EB/OL]. [2020-12-01]. https://www.software.ac.uk/blog/2018-06-22-measuring-behaviour-2018.

[23] HANNAY J E, MACLEOD C, SINGER J, et al. How do scientists develop and use scientific software? [C]//Proceedings of the 2009 ICSE workshop on software engineering for computational science and engineering. Piscataway: IEEE Computer Society, 2009: 1-8.

[24] PRABHU P, JABLIN T B, RAMAN A, et al. A survey of the practice of computational science[EB/OL]. [2020-12-15]. https://www.cs.princeton.edu/~dpw/papers/survey-sc11.pdf.

[25] HOWISON J, HERBSLEB J. The sustainability of scientific software: ecosystem context and science policy[EB/OL]. [2020-12-15]. http://james.howison.name/pubs/HowisonHerbsleb-Sustainability.pdf.

[26] SMITH A M, NIEMEYER K E, KATZ D S, et al. Design and first-year review[J/OL]. Journal of open source software[2021-05-20]. https://peerj.com/articles/cs-147.pdf.

[27] CHAWLA D S. The unsung heroes of scientific software[J]. Nature, 2016, 529(7584): 115-116.

[28] HWANG L, FISH A, SOITO L, et al. Software and the scientist: coding and citation practices in geodynamics [J]. Earth and space science, 2017, 4(11): 670-680.

[29] RIOS F. Preserving and sharing software for transparent and reproducible research: a review [EB/OL]. [2020-12-20]. https://osf.io/d4kef/.

[30] PARK H, WOLFRAM D. Research software citation in the Data Citation Index: current practices and implications for research software sharing and reuse [J]. Journal of informetrics, 2019(13): 574-582.

[31] HOWISON J, HERBSLEB J D. Scientific software production: incentives and collaboration [C/OL]. [2020-12-10]. https://dl.acm.org/doi/10.1145/1958824.1958904.

[32] HWANG L J, PAULOO R A, CARLEN J. Assessing the impact of outreach through software citation for community software in geodynamics [J]. Computing in science and engineering, 2020, 22(1): 16-25.

[33] LI K, CHEN P-Y, YAN E. Challenges of measuring software impact through citations: an examination of the lme4 R package [J]. Journal of informetrics, 2019, 13(1): 449-461.

[34] 史雅莉. 科学数据引用标准实施的关键问题探析[J]. 现代情报, 2019, 39(4): 34-41.

[35] NIEMEYER K E, SMITH A M, KATZ D S. The challenge and promise of software citation for credit, identification, discovery, and reuse[J]. Journal of data and information quality, 2016, 7(4): e16.

[36] DING Y, SONG M, HAN J, et al. Measuring the impact of entities [J]. PLOS one, 2013, 8(8): e71416.

[37] DING Y, ZHANG G, CHAMBERS T, et al. Content-based citation analysis: the next generation of citation analysis[J]. Journal of the Association for Information Science and Technology, 2014, 65(9): 1820-1833.

[38] YU Q, DING Y, SONG M, et al. Tracing database usage: Detecting main paths in database link networks [J]. Journal of informetrics, 2015, 9(1): 1-15.

[39] BARTZAS A, PEON-QUIROS M, POUCET C, et al. Software metadata: systematic characterization of the memory behaviour of dynamic applications [J]. Journal of systems and software, 2010, 83(6): 1051-1075.

[40] SOCIAS S M, MORIN A, TIMONY M A, et al. AppCiter: a web application for increasing rates and accuracy of scientific software citation [J]. Structure, 2015, 23(5): 807-808.

[41] THE CODEMETA PROJECT [EB/OL]. [2020-12-26]. https://codemeta.github.io/index.html.

[42] COSMO R D, GRUENPETER M, ZACCHIROLI S. Referencing source code artifacts: a separate concern in software citation [J]. Computing in science and engineering, 2020, 22(2): 33-43.

[43] 崔佳伟, 吴思竹, 邬金鸣, 等. 科学数据仓储元数据标准研究与启示[J]. 数字图书馆论坛, 2019(6): 19-28.

[44] TENOPIR C, ALLARD S, DOUGLASS K, et al. Data sharing by scientists: practices and perceptions [J]. PLos one, 2011, 6(6): 1-21.

[45] 浦燕妮, 刘琪, 耿骞. 通用型科学元数据标准研究[J]. 数字图书馆论坛, 2016(12): 33-39.

[46] KATZ D S, BOUQUIN D, CHUE HONG N P, et al. Software citation implementation challenges [EB/OL]. [2020-12-30]. https://arxiv.org/abs/1905.08674.

[47] DRUSKAT S. Software and dependencies in research citation graphs [J]. Computing in science & engineering, 2020, 22(2): 8-21.

[48] HENNEKEN E, DEVELOPER A. Asclepias - Enabling software citation & discovery workflows [EB/OL]. [2020-04-06]. https://adsabs.github.io/blog/asclepias.

[49] HENNEKEN E, ACCOMAZZI A, BLANCO-CUARESMA S, et al. Asclepias-Capturing software citations in astronomy[EB/OL]. [2020-10-08]. http://doi.org/10.5281/zenodo.1011088.

[50] HENNEKEN E A, ADS T. Capturing software citations in astronomy and planetary sciences [EB/OL]. [2020-10-09]. https://www.hou.usra.edu/meetings/lpsc2019/pdf/1569.pdf.

[51] ALLEN A, BERRIMAN G B, DUPRIE K, et al. Improving software citation and credit [EB/OL]. [2020-10-20]. https://arxiv.org/abs/1512.07919.

[52] 盛小平, 王毅. 利益相关者在科学数据开放共享中的责任与作用——基于国际组织科学数据开放共享政策的分析[J]. 图书情报工作, 2019, 63(17): 31-39.

[53] 张丽丽，黎建辉．数据引用的相关利益者分析[J]．情报理论与实践，2014(7)：44－47.
[54] 孟祥保，高凡．利益相关者视角下科研数据战略规划研究[J]．图书情报工作，2016，60(9)：38－44.
[55] SMITH A M，KATZ D S，NIEMEYER K E，et al. Software citation principles [J]. PeerJ Computer Science，2016，2：e86. https://doi.org/10.7717/peerj-cs.86.
[56] 完颜邓邓．国外科学数据仓储元数据实践调查及启示[J]．新世纪图书馆，2016(5)：81－84.
[57] 司莉，贾欢．科学数据的标准规范体系框架研究[J]．图书馆，2016(5)：5－9.
[58] 刘峰，张晓林．科学数据元数据标准述评及其通用化设计研究[J]．现代图书情报技术，2015(12)：3－12.
[59] 梁冰，王莉．关于国家科技管理信息系统标准体系建设的思考[J]．情报学报，2016，35(9)：911－916.
[60] 司莉，赵洁．美国开放政府数据元数据标准及启示[J]．图书情报工作，2018，62(3)：86－93.
[61] HSU C N，BANDROWSKI A E，GILLESPIE T H，et al. Comparing the use of research resource identi? ers and natural language processing for citation of databases，software，and other digital artifacts [J]. Computing in science & engineering，2020，22(2)：22－32.
[62] BALL A，DUKE M. 'How to track the impact of research data with metrics'. DCC how-to guides [EB/OL]. [2020－12－15]. https://www.dcc.ac.uk/guidance/how-guides/track-data-impact-metrics.
[63] DRUSKAT S. Citation file format -core module (CFF-Core) [EB/OL]. [2020－11－20]. https://zenodo.org/record/1108269#.XsibSfZuLIU.
[64] Working towards sustainable software for science: practice and experiences. Proceedings [EB/OL]. [2020－11－26]. http://wssspe.researchcomputing.org.uk/proceedings/.
[65] FORCE 11. Attribution working group [EB/OL]. [2020－11－26]. https://www.force11.org/group/attributionwg.
[66] FORCE 11. FORCE11-RDA joint software source code identification working group [EB/OL]. [2020－11－26]. https://www.force11.org/group/force11-rda-joint-software-source-code-identification-working-group.
[67] FORCE 11. Software citation working group [EB/OL]. [2020－11－26]. https://www.force11.org/group/software-citation-working-group.
[68] FORCE 11. Software citation implementation working group [EB/OL]. [2020－11－26]. https://www.force11.org/group/software-citation-implementation-working-group.
[69] Software Sustainability Institute. Other workshops [EB/OL]. [2020－11－26]. https://www.software.ac.uk/programmes-and-events/other-workshops.
[70] American Astronomical Society. Policy statement on software [EB/OL]. [2020－11－26]. https://journals.aas.org/policy-statement-on-software/.
[71] SWMATH [EB/OL]. [2020－11－26]. http://www.swmath.org/.
[72] HSL. A collection of Fortran codes for large scale scientific computation [EB/OL]. [2020－11－26]. http://www.hsl.rl.ac.uk/.
[73] WHITE O，DHAR A，BONAZZI V，et al. NIH software discovery index meeting report [R/OL]. [2020－11－20]. http://www.softwarediscoveryindex.org/.
[74] The journal of open source software [EB/OL]. [2020－11－26]. https://joss.theoj.org/.
[75] The journal of open research software [EB/OL]. [2020－11－26]. https://openresearchsoftware.metajnl.com/j.
[76] SOFTWAREX [EB/OL]. [2020－11－26]. https://www.journals.elsevier.com/softwarex/.
[77] THE CODEMETA PROJECT [EB/OL]. [2020－11－26]. https://codemeta.github.io/index.html.
[78] SHOULDACITE [EB/OL]. [2020－11－23]. https://mr-c.github.io/shouldacite/.
[79] COLLABORATIONS WORKSHOP 16 [EB/OL]. [2020－11－24]. https://www.software.ac.uk/cw16.
[80] DEPSY [EB/OL]. [2020－11－26]. https://github.com/ourresearch/depsy.
[81] IMPACTSTORY [EB/OL]. [2020－11－26]. https://github.com/ourresearch/impactstory-tng.
[82] GITHUB [EB/OL]. [2020－11－26]. https://github.com/.
[83] DATACITE [EB/OL]. [2020－11－26]. https://datacite.org/.
[84] GIL Y，RATNAKAR V，GARIJO D. OntoSoft: capturing scientific software metadata [C/OL]. [2020－11－30]. https://dl.acm.org/doi/10.1145/2815833.2816955.
[85] CITEAS [EB/OL]. [2020－11－29]. http://www.citeas.org/.
[86] HEIBI I，PERONI S，SHOTTON D. Software review: COCI，the open citations index of Crossref open DOI-to-DOI citations [J]. Scientometrics，2019，121(2)：1213－1228.
[87] MOZILLA SCIENCE LAB. Code as a rresearch object [EB/OL]. [2020－11－26]. http://mozillascience.github.io/code-research-object/.
[88] Software Carpentry [EB/OL]. [2021－01－03]. https://software-carpentry.org/.
[89] WILSON G. Software carpentry: lessons learned [version 2; peer review: 3 approved] [EB/OL]. [2020－11－18]. https://doi.org/10.12688/f1000research.3-62.v2.
[90] Training for Centres for Doctoral Training [EB/OL]. [2021－01－03]. https://www.software.ac.uk/programmes-and-events/carpentry-programmes/tailored-carpentry-workshops/training-centres-doctoral.
[91] The Engineering and Physical Sciences Research Council [EB/OL]. [2021－01－03]. https://epsrc.ukri.org/.

[92] CASRAI. CRediT[EB/OL]. [2021 -01 -03]. https://casrai.org/credit/.

[93] HOWISON J, HERBSLEB J D. Incentives and integration in scientific software production [C]//http://dio.org/10.1145/2441776.2441828.

[94] GRANT R. Life after my Fellowship [EB/OL]. [2021 - 01 - 02]. https://www.software.ac.uk/blog/2018-12-19-life-after-my-fellowship.

[95] ARNOLD B. Christopher Woods: how to design and engineer good code for research [EB/OL]. [2021 - 01 - 02]. https://www.software.ac.uk/blog/2019-03-11-christopher-woods-how-design-and-engineer-good-code-research.

[96] Society OF Research Software Engineering. RSE Fellows [EB/OL]. [2021 -01 -04]. https://society-rse.org/community/rse-fellows/.

[97] LAIRD J. Applications open for Better Scientific Software (BSSw) Fellowship Program [EB/OL]. [2021 -01 -02]. https://www.software.ac.uk/news/applications-open-better-scientific-software-bssw-fellowship-program.

[98] ARAGON S. US applicants sought for the 2019 Better Scientific Software (BSSw) Fellowship Programme[EB/OL]. [2021 -01 - 02]. https://www.software.ac.uk/news/us-applicants-sought-2019-better-scientific-software-bssw-fellowship-programme.

[99] WILSON R. Encouraging citation of software - introducing CITATION files [EB/OL]. [2020 -12 -25]. https://www.software.ac.uk/blog/2016-10-06-encouraging-citation-software-introducing-citation-files.

[100] DRUSKAT S, SPAAKS J H, STRUCK A. Hacking software citation implementation: the citation file format hack day at RSE18 [EB/OL]. [2021 - 12 - 27]. https://www.software.ac.uk/blog/2018-10-02-hacking-software-citation-implementation-citation-file-format-hack-day-rse18.

[101] DRUSKAT S. Track 2 lightning talk: should CITATION files be standardized? [EB/OL]. [2020 - 12 - 27]. https://figshare.com/articles/journal_contribution/Track_2_Lightning_Talk_Should_CITATION_files_be_standardized_/3827058.

[102] citation-file-format/citation-file-format [EB/OL]. [2020 - 12 - 28]. https://github.com/citation-file-format/citation-file-format#tools.

[103] SPAAKS J H, VERHOEVEN S, KLAVER T, et al. The research software directory and how it promotes software citation - Improve the findability, citability, and reproducibility of research software [EB/OL]. [2020 - 12 - 29]. https://www.software.ac.uk/blog/2018-12-17-research-software-directory-and-how-it-promotes-software-citation-improve.

作者贡献说明:
丁敬达:论文选题、研究思路和框架制定、论文撰写;
郑巧:研究思路和框架制定、论文撰写;
刘超:论文撰写。

Theoretical Exploration and Practical Progress of Software Citation and Its Specification

Ding Jingda Zheng Qiao Liu Chao
Department of Library, Information and Archives, Shanghai University, Shanghai 200444

Abstract: [**Purpose/significance**] This paper surveys the status of theory and practice of software citation, analyzes the existing difficulties and problems, and promotes the establishment of software citation norm and standard. [**Method/process**] Through the literature research, it's found that software citation has faced four challenges: acceptance culture, reward system, citation cognition and metadata. Then this paper analyzed the exploration of theory and practice of software citation and its specification at home and abroad in order to meet the challenges. [**Result/conclusion**] Stakeholder analysis of software citation, theoretical exploration of software citation principles and metadata standards, citation guidelines of international research organizations and communities, support of related projects and databases, personnel training, implementation of contribution distribution and reward plan, and provision of software citation documents have laid a good foundation for the establishment of software citation norms, but the stakeholders still need work together to overcome the difficulties and challenges of software citation.

Keywords: software citation software citation principle software citation norm software citation practice